week_6

krishna sai surendra babu kalluri

17/07/2020

1)Bagging: Bootstrap aggregation or bagging is a procedure that is used to reduce variance for a statistical method.Generally, the decision trees will suffer with high variance , when we split the data into training data it gives different values. so bagging is used to reduce the variance by averaging a set of observations.

To obtain average prediction we have the formula

$$f_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B} f^{b}\,(x)$$

Advantages:
1)These trees will grow deeply without any pruning 2)It helps avoiding overfitting

Disadvantages:

1)loss of interpretation

Random forest :

Random forest offers an development over bagged trees by random small tweak that decorrelates trees.

Unlike Bagging, Instead of splitting tree everytime, a random sample of m predictors will choose from a set of p predictors and spliting is allowed only one of these m predictors and we choose nearly

$$m = \sqrt{p}$$

The number predictors at each split is equal to square root of total number of available predictors

Explaination: Let suppose if we have some sample of predictors and among them there is one strong predictor with average strong predictors ,

Bagging used to consider all the predictors ,so averaging these predictors does not reduce much variance.This can be overcome by Random forest by splitting only set of predictors from full predictors.

Advantages: 1)It can solve both classification and regression 2)It can handle large data sets with high dimensionality .

3)It is an effective method for estimating missing data and maintains accuracy

Disadvantages:

1)complexity

2)Long time period

Boosting:

Boosting is a sequential process ,where next tree gets the information from the previous trees and tries to accurate the mistakes from the previous tree.

Explaination: Boosting is similar to bagging where we create multiple copies for the original data set using bootstrap and splitting the trees ,fitting separate decision tree for each copy and combining all of the trees to create a predictive model.But in boosting the trees grows sequentially ,each tree will get the information from the previous tree . Here ,we fit the decision tree using current residuals and adding the new decision tree into fitted function in order to update the residuals so that each tree will improves that are not performing well and all the trees will grow stronger. To obtain the boosting model we have formula

$$f(x) = \sum_{b=1}^{B} \lambda f^b(X)$$

Advantages:

1)It is easy to read and interpret algorithm and easy to handle. 2)It can make weak learners to strong learners

Disadvatages:

1)Time complexity for larger data sets 2)Method is almost impossible to scale up because every tree bases it correctness on the previous tree which makes procedure to streamline

Here ,From the below code we compared the Bagging,boosting and random forest .Hence , we can conclude that no one is greater than other .It will choose depends upon the situation.

```
knitr::opts_chunk$set(echo = TRUE)
library( randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.0.2
```

```
## randomForest 4.6-14
```

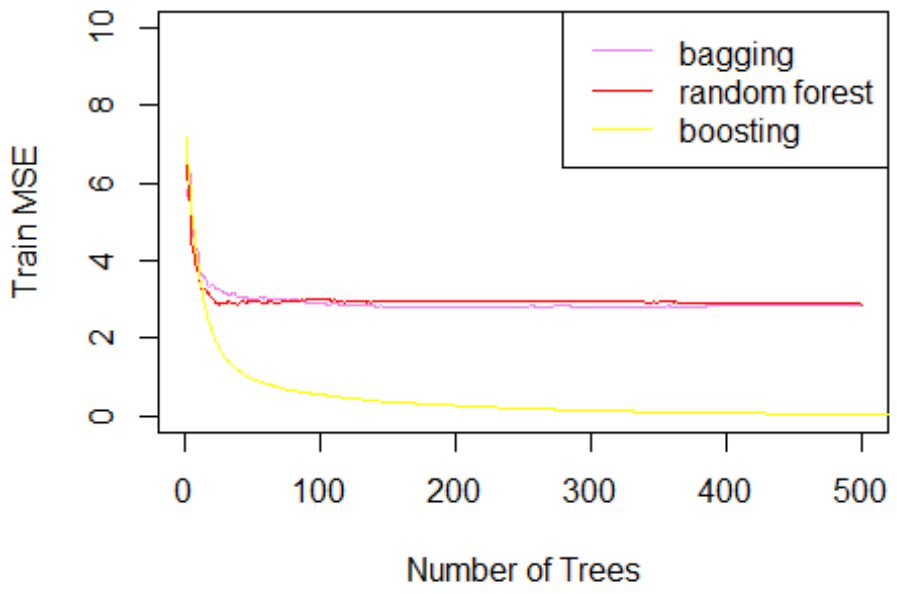
```
## Type rfNews() to see new features/changes/bug fixes.
```

```
library(ISLR)
library(gbm)
```

```
## Warning: package 'gbm' was built under R version 4.0.2
```

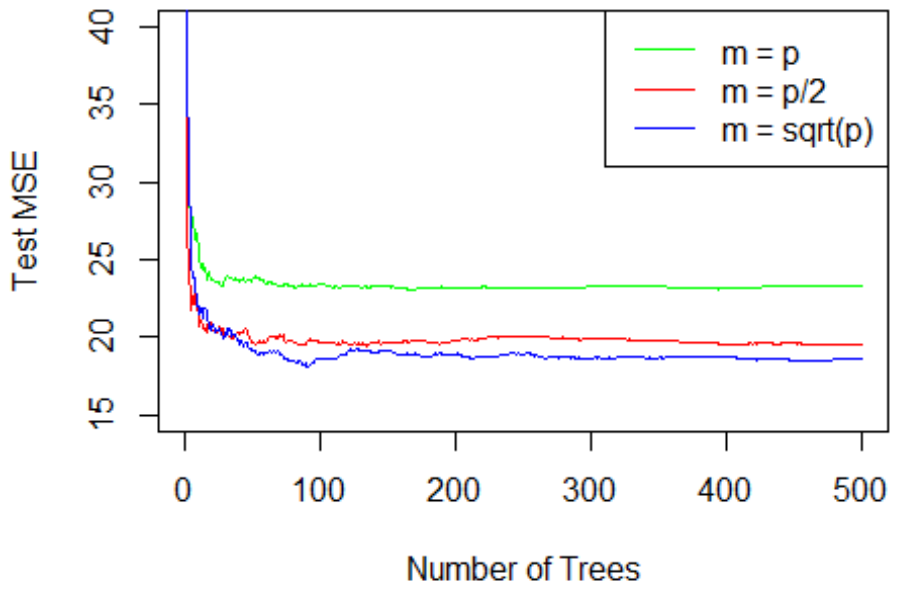
```
## Loaded gbm 2.1.8
```

```
set.seed(1)
train = sample (1:nrow(Carseats), nrow(Carseats)/2)
Carseats.train=Carseats[train,-10]
Carseats.test=Carseats[-train ,-10]
cs.train <- Carseats[train,10]
cs.test <- Carseats[-train,10]
bag.Carseats = randomForest(Sales~ ., data = Carseats, subset = train,
mtry=13)

## Warning in randomForest.default(m, y, ...): invalid mtry: reset to within
valid
## range

rf.Carseats= randomForest(Sales~.,data=Carseats , subset=train , mtry=6)

bos.Carseats = gbm(Sales~.,data=Carseats[train ,], distribution=
"gaussian",n.trees=5000, interaction.depth=4)

plot(1:500, bag.Carseats$mse, col = "violet", type = "l", xlab = "Number of
Trees", ylab = "Train MSE", ylim = c(0,10))
lines(1:500, rf.Carseats$mse, col = "red", type = "l")
lines(1:5000, bos.Carseats$train.error, col = "yellow", type = "l")

legend("topright", c("bagging", "random forest", "boosting"), col =
c("violet", "red", "yellow"), cex = 1, lty = 1)
```



2 a)

```
library( randomForest)
library(MASS)
set.seed(1)
train = sample (1:nrow(Boston), nrow(Boston)/2)
boston.train=Boston[train,-14]
boston.test=Boston[-train ,-14]
bos.train <- Boston[train,14]
bos.test <- Boston[-train,14]
bag.boston1 = randomForest(boston.train, y=bos.train,xt=boston.test,yt=
bos.test,mtry=ncol(Boston) - 1,ntree = 500)
bag.boston2 = randomForest(boston.train, y=bos.train,xt=boston.test,yt=
bos.test ,mtry=(ncol(Boston) - 1) / 2,ntree = 500)
bag.boston3 = randomForest(boston.train, y=bos.train,xt=boston.test,yt=
bos.test ,mtry=sqrt(ncol(Boston) - 1),ntree = 500)
plot(1:500, bag.boston1$test$mse, col = "green", type = "l", xlab = "Number
of Trees", ylab = "Test MSE", ylim = c(15,40))
lines(1:500, bag.boston2$test$mse, col = "red", type = "l")
lines(1:500, bag.boston3$test$mse, col = "blue", type = "l")

legend("topright", c("m = p", "m = p/2", "m = sqrt(p)"), col = c("green",
"red", "blue"), cex = 1, lty = 1)
```




```
knitr::opts_chunk$set(echo = TRUE)
```

2 b)

```
importance(bag.boston1)
```

```
##         IncNodePurity
## crim        799.49055
## zn           71.16559
## indus        97.51018
## chas         10.66954
## nox         257.36785
## rm        12421.10663
## age         311.36525
## dis         257.24794
## rad          65.71323
## tax         137.94026
## ptratio     126.68287
## black       223.72489
## lstat      4682.99582
```

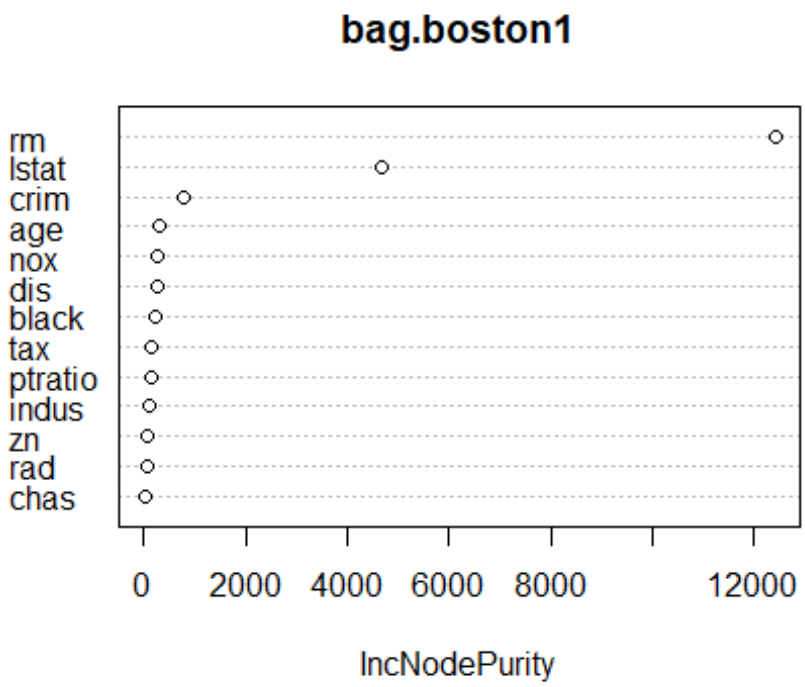
```
importance(bag.boston2)
```

```
##         IncNodePurity
## crim       1062.88179
## zn           90.35011
## indus       604.38052
## chas         44.04921
## nox         750.40873
## rm         7733.73810
## age         598.32370
## dis         764.53866
## rad          88.05417
## tax         315.19230
## ptratio     902.27030
## black       272.48096
## lstat      6145.75065
```

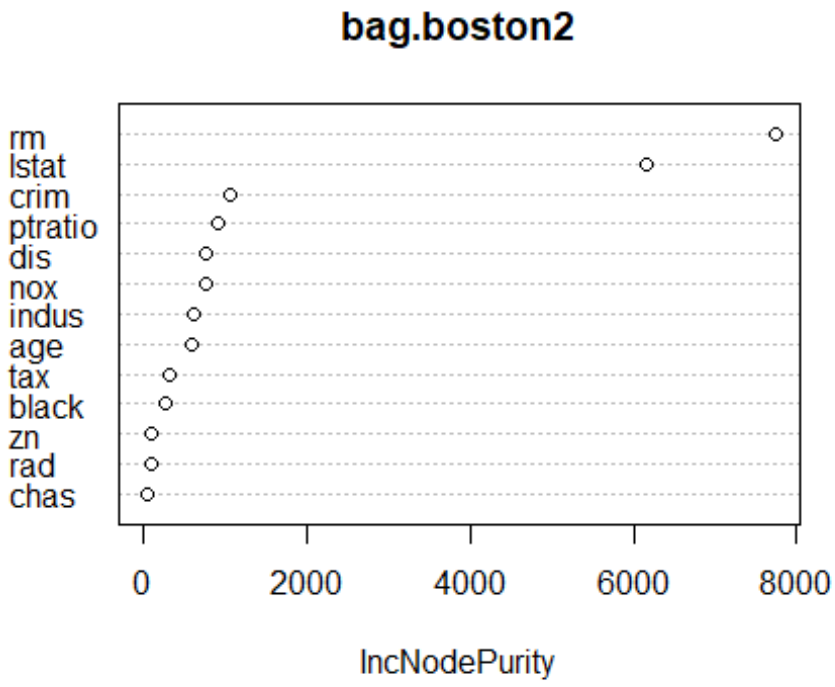
```
importance(bag.boston3)
```

```
##         IncNodePurity
## crim       1298.26055
## zn          198.51130
## indus       937.04932
## chas         58.37771
## nox        1109.13687
## rm         5836.30309
## age         819.90156
## dis         848.93018
## rad         148.45790
## tax         607.96120
## ptratio    1161.50794
## black       337.25694
## lstat      5497.42287
```

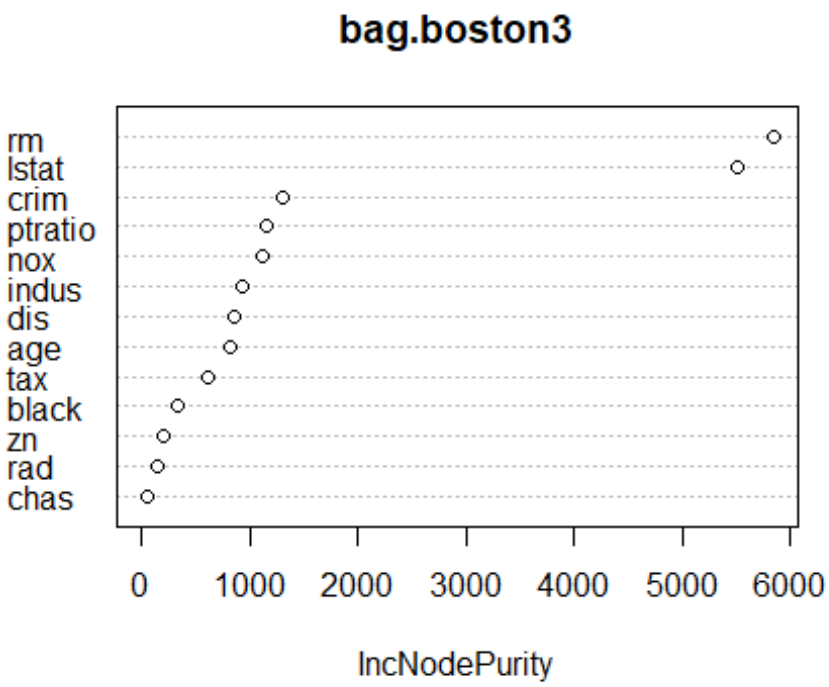
```
varImpPlot (bag.boston1)
```




```
varImpPlot(bag.boston2)
```




```
varImpPlot(bag.boston3)
```




```
knitr::opts_chunk$set(echo = TRUE)
```

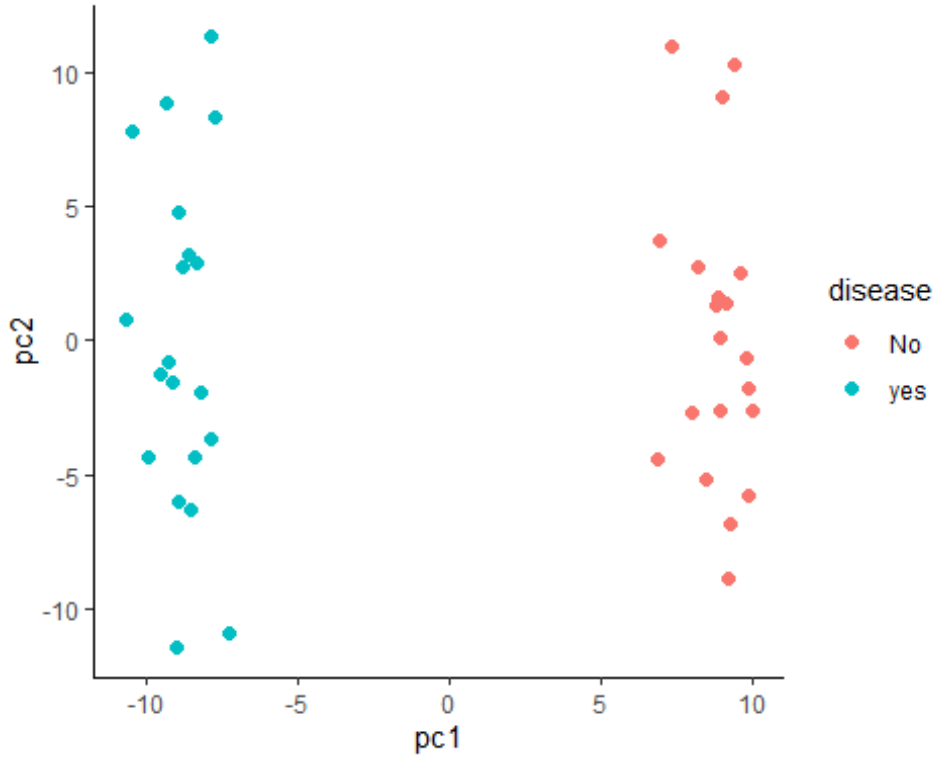
b) From the above code ,we can conclude that lstat(level of community) and rm(house size) are the most important variable.yes important variables differ for different values of mtry.

Questions from Thursday:

3 a)

```
knitr::opts_chunk$set(echo = TRUE)
## 3 a)

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.0.2

##
## Attaching package: 'ggplot2'

## The following object is masked from 'package:randomForest':
##
##     margin

data <- read.csv("c:\\Users\\kallu\\Ch10Ex11.csv",header = FALSE)
genes <- t(data)
data.pca=prcomp(genes , scale=TRUE)
df <- cbind.data.frame(data.pca$x[,1],data.pca$x[,2])
```

```
df$disease <-c(rep("yes",20),rep("No",20))
colnames(df)[1:3] <-c("pc1","pc2",'disease')



ggplot(df,aes(x=pc1,y=pc2,col=disease))+geom_point(size=2,alpha=1)+theme_clas
sic()
```



3) b)From the results ,we can conclude that health tissue differs more than diseased tissue .As we can see from the plot X08188148 are seperate from the group which is a health tissue .So we can conclude that health tissue differs more

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

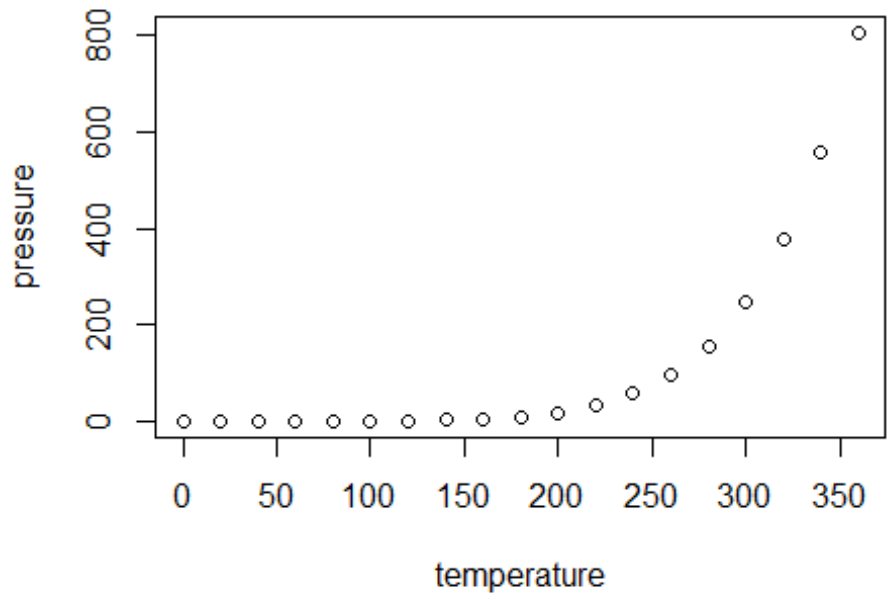
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)

##      speed           dist
##  Min.   : 4.0   Min.   :  2.00
```

```
##  1st Qu.:12.0   1st Qu.: 26.00
##  Median :15.0   Median : 36.00
##  Mean   :15.4   Mean   : 42.98
##  3rd Qu.:19.0   3rd Qu.: 56.00
##  Max.   :25.0   Max.   :120.00
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.